Lab 1

San Francisco State University

ENGR 476-01 Computer Communication Networks
Spring 2023

Christie Lai

Date of submission: 2/14/2023

About the Lab:

Lab 1 is designed to help students refresh their coding skills with C. This coding assignments required us to write a file in C and read through the file. Once the file is read, we had to get users interaction with the code and the file by prompting several commands. When the command is executed then the code is run and it will output what the users is looking for.

Code:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <math.h>

#define SSIZE 1024 // size of string

// Declarations
int read = 0;
int columns = 0;
char first_line[100];
int rows = 7;
float length = 7;
char current3[100];

typedef struct
{
    char student[10];
    int student_no;
    float test_score1;
    float test_score2;
    char row[100];

} students_record;

students_record array[7]; // can store up to 10 student records into the array

FILE *fp;
FILE *fp1;
char input; // command inputs
char string[SSIZE];
float test1;
float test2;
char *students_names;
```

```
char file_name[100], c;

void prompt() // Function for prompt when h is inputed
{

    printf("a/1 to obtain all the students that got diploma \n"
           "b/2 to arrange subject A in ascending order \n"
           "c/3 to calculate the average and standard deviation \n"
           "d/4 to save all the above results in an output file \n"
           "e to exit \n");
}

void diplomas(students_record array[]) // Function to calculate the students that
got diplomas
{

    while (fgets(string, SSIZE, fp) != NULL) // read thorugh each line in the
text file
    {
        char *token = strtok(string, " "); // break string into series of tokens

        token = strtok(NULL, " "); // returns a NULL pointer if there is no
strings
        students_names = token;    // makes names into a token

        token = strtok(NULL, " "); // convert token to floating point value
        test1 = atof(token);       // input values in to data type float test1

        token = strtok(NULL, " ");
        test2 = atof(token);

        if (test1 >= 50 && test2 >= 50) // logic to find students with diplomas
with the score >= 50
        {
            printf("%-15s  %-14g  %g \n", string, test1, test2);
            fprintf(fp1, "%-15s  %-14g  %g \n", string, test1, test2);
        }
    }
}

void AscendingOrder(students_record array[]) // Function to put Subject A in
ascending order
{
    do
    {
```

```
        // gets the frist line of the file and moves on to the next one
        fgets(first_line, 100, fp);

        // read a line from the file
        read = fscanf(fp, "%s %d %g %g", array[columns].student,
                      &array[columns].student_no,
                      &array[columns].test_score1,
                      &array[columns].test_score2);

        if (read == 4) // read is successful if the number of read matches the
number of columns
            columns++;

        if (read != 4 && !feof(fp)) // format file error
        {
            printf("File formaat is incorrect.\n");
            exit(0);
        }

    } while (!feof(fp));

    // continue if we have not reach end of the file
    // Use selection sort Subject A in asccending order
    // go through array one element at a time up to the last element
    for (int i = 0; i < length - 1; i++)
    {
        // find the minimum position in the array
        int min = i;
        for (int j = i + 1; j < length; j++) // find number smaller then the min
position number

            if (array[j].test_score1 < array[min].test_score1)
                min = j;

        if (min != i) // preform swap with the min position number with with
current number
        {
            float current = array[i].test_score1;
            array[i].test_score1 = array[min].test_score1;
            array[min].test_score1 = current;

            int current2 = array[i].student_no;
            array[i].student_no = array[min].student_no;
            array[min].student_no = current2;
```

```
            float current1 = array[i].test_score2;
            array[i].test_score2 = array[min].test_score2;
            array[min].test_score2 = current1;

            strcpy(current3, array[i].student);
            strcpy(array[i].student, array[min].student);
            strcpy(array[min].student, current3);
        }
    }
    // loop to go through and read each line of the file store the record into
the student array
    for (int i = 0; i < columns; i++) // print out the array of lines that has
been read in the file

    {
        printf("%-16s  %-14d  %-14g  %g \n", array[i].student,
               array[i].student_no,
               array[i].test_score1,
               array[i].test_score2);

        fprintf(fp1, "%-16s  %-14d  %-14g  %g \n", array[i].student,
                array[i].student_no,
                array[i].test_score1,
                array[i].test_score2);
    }
}

void Calculations() // Function to calculate the average and standard deviation
of Subject A and Subjuct B
{
    float meanA, meanB, standard_deviationA, standard_deviationB;
    float sumA = 0, sumB = 0;

    for (int i = 0; i < length; i++) // loops that sum
    {
        size_t length = sizeof(array[i].test_score1); // size of the arrray
        sumA = sumA + array[i].test_score1;
        sumB = sumB + array[i].test_score2;
    }
    meanA = sumA / length;
    meanB = sumB / length;

    standard_deviationA = meanA / sqrt(length);
    standard_deviationB = meanB / sqrt(length);
```

```
    printf("Subject A Average: %g\n"
           "Subject B Average: %g\n",
           meanA, meanB);

    printf("\nSubject A Standard Deviation: %g\n"
           "Subject B Standard Deviation: %g\n",
           standard_deviationA, standard_deviationB);

    fprintf(fp1, "Subject A Average: %g\n"
                 "Subject B Average: %g\n",
            meanA, meanB);

    fprintf(fp1, "\nSubject A Standard Deviation: %g\n"
                 "Subject B Standard Deviation: %g\n",
            standard_deviationA, standard_deviationB);
}

int main()
{
    // Prompt the user to enter a file to read and output the content of the file
    printf("Please enter input file name: ");
    gets(file_name);

    fp = fopen(file_name, "r"); // opens the file and read the content data

    c = fgetc(fp);
    while (c != EOF)
    {
        printf("%c", c);
        c = fgetc(fp);
    }

    if (fp == NULL)
    {                                       // check to see if the file can open
        printf("Error opening the file.\n"); // if file does not open it
terminates the file
        return 1;
    }

    do
    {
        // asks prompt
        printf("\nPlease enter a command (enter h for help): ");
        scanf(" %c", &input);
```

```
        fp = fopen(file_name, "r");

        // when h is inputted then it outputs the prompt() above
        if (input == 'h')
        {
            prompt();
        }

        else if (input == 'a' || input == '1')
        {

            printf("All the students that got diplomas:\n"
                   "STUDENT NAME \t SUBJECT A \t SUBJECT B \n");
            diplomas(array);
        }

        else if (input == 'b' || input == '2')
        {

            printf("Subject A in ascending order:\n"
                   "STUDENT NAME \t STUDENT NO. \t SUBJECT A \t SUBJECT B \n");
            printf("Subject A in ascending order:\n"
                   "STUDENT NAME \t STUDENT NO. \t SUBJECT A \t SUBJECT B \n");

            // loop to go through and read each line of the file store the record
into the student arra
            do
            {
                // gets the frist line of the file and moves on to the next one
                fgets(first_line, 100, fp);

                // read a line from the file
                read = fscanf(fp, "%s %d %g %g", array[columns].student,
                              &array[columns].student_no,
                              &array[columns].test_score1,
                              &array[columns].test_score2);

                if (read == 4) // read is successful if the number of read
matches the number of columns
                    columns++;

                if (read != 4 && !feof(fp)) // format file error
                {
                    printf("File formaat is incorrect.\n");
                    return 1;
```

```
                }

            } while (!feof(fp));

            // continue if we have not reach end of the file
            // Use selection sort Subject A in asccending order
            // go through array one element at a time up to the last element
            for (int i = 0; i < length - 1; i++)
            {
                // find the minimum position in the array
                int min = i;
                for (int j = i + 1; j < length; j++) // find number smaller then
the min position number

                    if (array[j].test_score1 < array[min].test_score1)
                        min = j;

                if (min != i) // preform swap with the min position number with
with current number
                {
                    float current = array[i].test_score1;
                    array[i].test_score1 = array[min].test_score1;
                    array[min].test_score1 = current;

                    int current2 = array[i].student_no;
                    array[i].student_no = array[min].student_no;
                    array[min].student_no = current2;

                    float current1 = array[i].test_score2;
                    array[i].test_score2 = array[min].test_score2;
                    array[min].test_score2 = current1;

                    strcpy(current3, array[i].student);
                    strcpy(array[i].student, array[min].student);
                    strcpy(array[min].student, current3);
                }
            }

            for (int i = 0; i < columns; i++) // print out the array of lines
that has been read in the file

                printf("%-16s  %-14d  %-14g  %g \n", array[i].student,
                       array[i].student_no,
                       array[i].test_score1,
                       array[i].test_score2);
```

```
        }

        else if (input == 'c' || input == '3')
        {
            printf("The Average and Standard Deviation of each Subject: \n");

            float meanA, meanB, standard_deviationA, standard_deviationB;
            float sumA = 0, sumB = 0;

            for (int i = 0; i < length; i++)
            {
                size_t length = sizeof(array[i].test_score1); // size of the
arrray
                sumA = sumA + array[i].test_score1;
                sumB = sumB + array[i].test_score2;
            }
            meanA = sumA / length;
            meanB = sumB / length;

            standard_deviationA = meanA / sqrt(length);
            standard_deviationB = meanB / sqrt(length);

            printf("Subject A Average: %g\n"
                   "Subject B Average: %g\n",
                   meanA, meanB);

            printf("\nSubject A Standard Deviation: %g\n"
                   "Subject B Standard Deviation: %g\n",
                   standard_deviationA, standard_deviationB);
        }

        else if (input == 'd' || input == '4')
        {
            fp1 = fopen("Output.txt", "w+");

            if (fp == NULL)
            {                                       // check to see if the file
can open
                printf("Error opening the file.\n"); // if file does not open it
terminates the file
            }

            fprintf(fp1, "All the students that got diplomas:\n"
                         "STUDENT NAME \t SUBJECT A \t SUBJECT B \n");
            printf("All the students that got diplomas:\n"
```

```
                "STUDENT NAME \t SUBJECT A \t SUBJECT B \n");
            diplomas(array);
            printf("\n");

            fprintf(fp1, "\nSubject A in ascending order:\n"
                         "STUDENT NAME \t STUDENT NO. \t SUBJECT A \t SUBJECT B
\n");
            printf("Subject A in ascending order:\n"
                "STUDENT NAME \t STUDENT NO. \t SUBJECT A \t SUBJECT B \n");
            AscendingOrder(array);
            printf("\n");

            fprintf(fp1, "\nStandard Deviation of each Subject: \n");
            printf("Standard Deviation of each Subject: \n");
            Calculations();

            fclose(fp1);
        }
    }

    while (input != 'e'); // exit program

    exit(0);

    fclose(fp); // close and saves the written data into the file */

    return 0;
}
```

data.txt:

C_C++ > Assignment_1 > data.txt

	STUDENT NAME	STUDENT NO.	SUBJECT A	SUBJECT B
2	JOAN	1	70.5	85
3	TANIA	2	49	75
4	TOM	3	53	54
5	JEFF	4	80	49.5
6	SUSAN	5	89	90
7	KATHY	6	99	55
8	RAYMOND	7	22.5	75
9				

Output.txt:

C_C++ > Assignment_1 > Output.txt

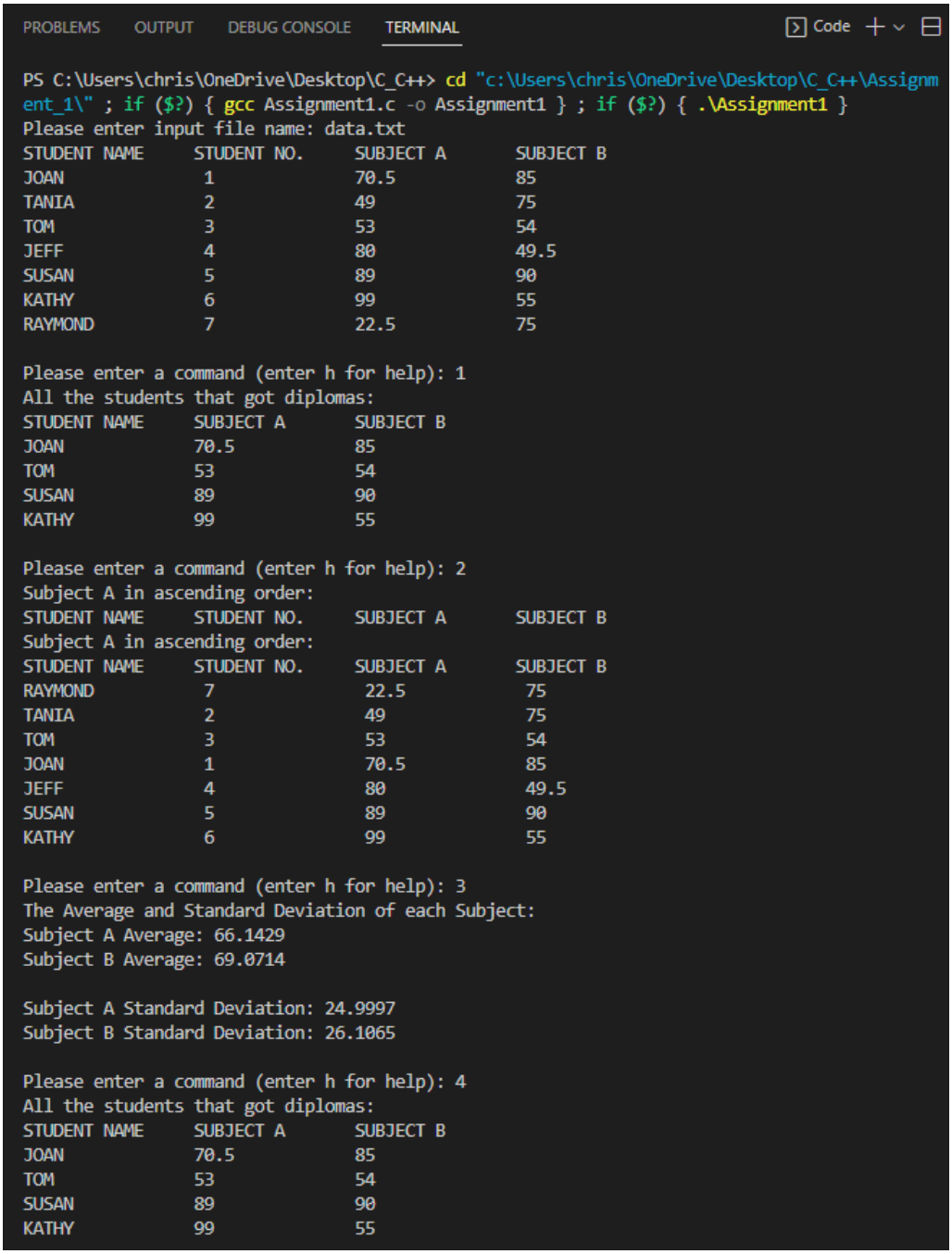
```
All the students that got diplomas:
STUDENT NAME     SUBJECT A   SUBJECT B
JOAN             70.5            85
TOM              53              54
SUSAN            89              90
KATHY            99              55

Subject A in ascending order:
STUDENT NAME     STUDENT NO.     SUBJECT A   SUBJECT B
RAYMOND           7               22.5            75
TANIA             2               49              75
TOM               3               53              54
JOAN              1               70.5            85
JEFF              4               80              49.5
SUSAN             5               89              90
KATHY             6               99              55

Standard Deviation of each Subject:
Subject A Average: 66.1429
Subject B Average: 69.0714

Subject A Standard Deviation: 24.9997
Subject B Standard Deviation: 26.1065
```

Output:



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL                                Code

PS C:\Users\chris\OneDrive\Desktop\C_C++> cd "c:\Users\chris\OneDrive\Desktop\C_C++\Assignm
ent_1\" ; if ($?) { gcc Assignment1.c -o Assignment1 } ; if ($?) { .\Assignment1 }
Please enter input file name: data.txt
STUDENT NAME     STUDENT NO.     SUBJECT A       SUBJECT B
JOAN              1              70.5            85
TANIA             2              49              75
TOM               3              53              54
JEFF              4              80              49.5
SUSAN             5              89              90
KATHY             6              99              55
RAYMOND           7              22.5            75

Please enter a command (enter h for help): 1
All the students that got diplomas:
STUDENT NAME     SUBJECT A       SUBJECT B
JOAN             70.5            85
TOM              53              54
SUSAN            89              90
KATHY            99              55

Please enter a command (enter h for help): 2
Subject A in ascending order:
STUDENT NAME     STUDENT NO.     SUBJECT A       SUBJECT B
Subject A in ascending order:
STUDENT NAME     STUDENT NO.     SUBJECT A       SUBJECT B
RAYMOND           7               22.5            75
TANIA             2               49              75
TOM               3               53              54
JOAN              1               70.5            85
JEFF              4               80              49.5
SUSAN             5               89              90
KATHY             6               99              55

Please enter a command (enter h for help): 3
The Average and Standard Deviation of each Subject:
Subject A Average: 66.1429
Subject B Average: 69.0714

Subject A Standard Deviation: 24.9997
Subject B Standard Deviation: 26.1065

Please enter a command (enter h for help): 4
All the students that got diplomas:
STUDENT NAME     SUBJECT A       SUBJECT B
JOAN             70.5            85
TOM              53              54
SUSAN            89              90
KATHY            99              55
```

```
Please enter a command (enter h for help): 4
All the students that got diplomas:
STUDENT NAME     SUBJECT A       SUBJECT B
JOAN             70.5            85
TOM              53              54
SUSAN            89              90
KATHY            99              55

Subject A in ascending order:
STUDENT NAME     STUDENT NO.     SUBJECT A       SUBJECT B
RAYMOND           7               22.5            75
TANIA             2               49              75
TOM               3               53              54
JOAN              1               70.5            85
JEFF              4               80              49.5
SUSAN             5               89              90
KATHY             6               99              55

Standard Deviation of each Subject:
Subject A Average: 66.1429
Subject B Average: 69.0714

Subject A Standard Deviation: 24.9997
Subject B Standard Deviation: 26.1065

Please enter a command (enter h for help): h
a/1 to obtain all the students that got diploma
b/2 to arrange subject A in ascending order
c/3 to calculate the average and standard deviation
d/4 to save all the above results in an output file
e to exit

Please enter a command (enter h for help): e
PS C:\Users\chris\OneDrive\Desktop\C_C++\Assignment_1>
```